Two-Stage LLM-as-a-Judge Using Required and Bonus Criteria

Kohei Tsunefuji[1] **Akihiro Tamura**[1] **Daichi Nakatsuji**[2]
Hiroaki Obayashi[2] **Mitsuo Hamada**[2] **Tsuneo Kato**[1]

Doshisha University[1] transcosmos inc.[2]
ctwn0178@mail4.doshisha.ac.jp
{aktamura, tsukato}@mail.doshisha.ac.jp
{Daichi.Nakatsuji, Hiroaki.Oobayashi, mitsuo.hamada}@transcosmos.com

Abstract

This study proposes a new large language model (LLM)-as-a-Judge method for the relative evaluation of LLM responses based on both required criteria (i.e., minimal conditions that responses must satisfy) and bonus criteria (i.e., desirable additional qualities). Recent LLM-as-a-Judge methods have improved the stability and accuracy of relative evaluation by employing automatically generated criteria. However, existing methods rely solely on required criteria, which may limit their ability to distinguish between closely matched responses. To address this issue, the proposed method first evaluates responses based on required criteria and, in the case of a tie, further evaluates them using bonus criteria, enabling finer-grained comparisons. Experiments conducted on the Writing, Roleplay, and Humanities categories of the MT-Bench dataset demonstrate that the proposed method achieves higher agreement with human judgments than existing methods.

1 Introduction

With recent advances in large language models (LLMs), the automatic evaluation of LLMs has become increasingly important. One approach to evaluating LLM performance involves comparing LLM responses with reference answers using automatic evaluation metrics (Brown et al., 2020; Wei et al., 2022), such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and BERTScore (Zhang et al., 2020). However, for open-ended generative tasks, such as email writing, where no single reference answer exists, these methods are insufficient for reliable performance evaluation.

To address this issue, Zheng et al. (2023) proposed LLM-as-a-Judge, in which a high-performing LLM (judge LLM) evaluates the responses of another LLM (target LLM). Both the original study and subsequent work, including Chiang and Lee (2023), have reported that LLM-as-a-Judge achieves evaluation performance comparable to that of human evaluators.

LLM-as-a-Judge can be broadly categorized into two types: absolute evaluation, which independently scores each response generated by a target LLM, and relative evaluation, which compares responses from two target LLMs to determine which performs better. In this work, following Chen et al. (2024), we focus on reference-free relative evaluation using LLM-as-a-Judge.

Early LLM-as-a-Judge methods suffer from issues such as inconsistency caused by unclear evaluation criteria and *position bias* (Wang et al., 2024), whereby evaluation results are influenced by the order in which responses are presented. To mitigate these issues, recent approaches have improved the stability and accuracy of relative evaluation by explicitly incorporating evaluation criteria into judge LLMs. For example, Kim et al. (2023, 2024) proposed Prometheus, an open-source judge LLM that performs evaluation based on user-defined criteria. However, because the criteria are provided externally, Prometheus does not dynamically generate evaluation criteria for each instruction. Qin et al. (2024) proposed INFOBENCH, a benchmark for evaluating LLM responses using manually constructed checklists with explicit evaluation criteria. Cook et al. (2024) proposed TICK, a method that automatically generates yes/no-style evaluation criteria and then compares LLM responses based on the proportion of satisfied criteria.

However, existing methods rely solely on required criteria and do not consider desirable qualities beyond these minimum requirements. For example, for the prompt "Write a travel blog article that includes must-see spots in Hawaii," responses are conventionally evaluated based on minimum requirements, such as whether the response is written in the form of a blog article and includes must-see spots in Hawaii. In contrast, considering additional desirable qualities, such as readability and practi-

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 768–777
November 6-10, 2026 ©2026 Association for Computational Linguistics

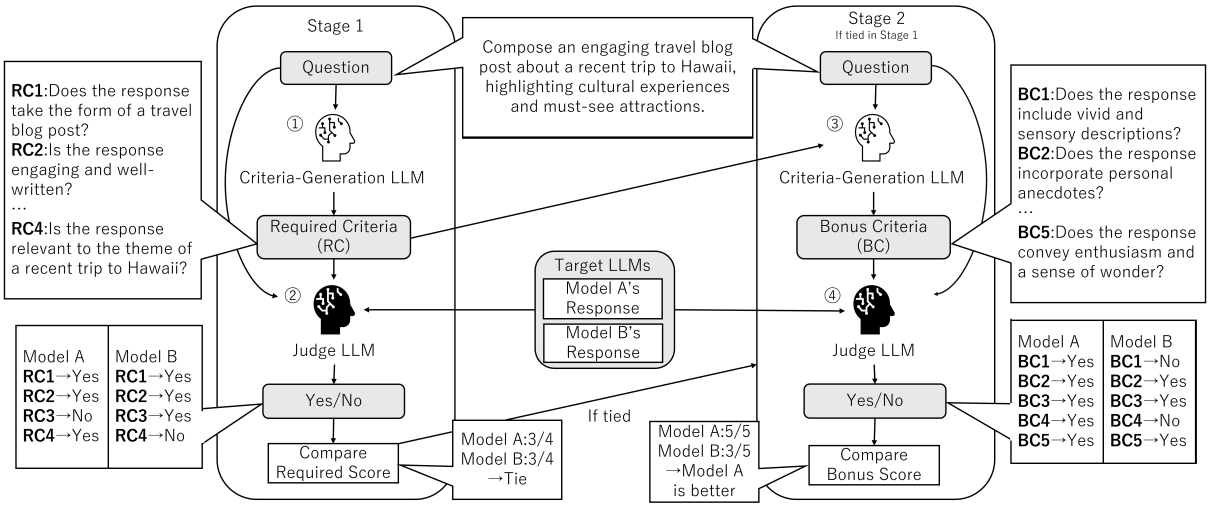


Figure 1: Overview of the Proposed Two-Stage LLM-as-a-Judge Method

cal tips, could enable finer-grained evaluation.

In this paper, we propose a two-stage LLM-as-a-Judge method that evaluates responses based on both required criteria and bonus criteria, which represent desirable qualities that responses are encouraged to satisfy. Specifically, in the first stage, the proposed method performs relative evaluation based on required criteria, similar to the method of Cook et al. (2024). When the evaluation results are tied, the second stage performs relative evaluation using bonus criteria, enabling finer-grained comparisons. Experiments on the MT-Bench dataset (Zheng et al., 2023) demonstrate that the proposed method achieves higher agreement with human evaluations than existing LLM-as-a-Judge methods. Furthermore, human evaluations confirm that both the criteria-generation LLM and the judge LLM perform accurately. In addition, we publicly release the prompts and human evaluation data used in this study to facilitate future research[1].

2 Two-Stage LLM-as-a-Judge

Figure 1 shows an overview of the proposed method. The proposed method is a two-stage framework. The first stage performs relative evaluation based on required criteria. In the case of a tie, the second stage further evaluates the responses using bonus criteria. In both stages, responses are evaluated criterion by criterion rather than holistically, and are compared based on the proportion of satisfied criteria. This criterion-level evaluation enables the identification of the aspects of the responses that contributed to the final judgment.

In the first stage, a criteria-generation LLM generates required criteria that responses should satisfy, based on the question. Subsequently, the question, the generated criteria, and the responses of the target LLMs are provided to a judge LLM, which evaluates each required criterion in a yes/no manner. A "yes" judgment indicates that the criterion is satisfied. The proportion of satisfied criteria is computed from the judge LLM's yes/no judgments and used as the required score. The prompts used for both the criteria-generation LLM and the judge LLM are based on those of Cook et al. (2024) and adapted to a multi-turn setting (Appendix B). A required score is computed for each target LLM, where a higher score indicates better performance. If a clear winner is determined at this stage, the evaluation concludes. Otherwise, when the required scores are equal, the second stage performs relative evaluation using bonus criteria.

In the second stage, the criteria-generation LLM receives the question and the required criteria generated in the first stage, and generates bonus criteria using a prompt that instructs it to produce desirable evaluation criteria that responses are encouraged to satisfy (Appendix B). The generated bonus criteria are formulated in a yes/no (satisfied / not satisfied) format, similar to the required criteria[2].

The question, the generated bonus criteria, and the responses of the target LLMs are provided to

[1]The prompts and human evaluation data are released under the CC BY 4.0 License at https://github.com/Tsunefuji/Two-Stage-LLM-as-a-Judge-paper.

[2]Following Cook et al. (2024), the number of required criteria is set to 2–8. In contrast, the number of bonus criteria is unconstrained, since the appropriate number varies substantially across tasks.

the judge LLM, which determines whether each bonus criterion is satisfied. Based on the yes/no judgments of the judge LLM, the proportion of satisfied bonus criteria is computed as the bonus score. A bonus score is computed for each target LLM, and the model with the higher score is considered superior; if the scores are identical, the result is treated as a tie. The required and bonus criteria for each question are shared across all target LLMs.

3 Experiments

3.1 Experimental Setup

We evaluated the effectiveness of the proposed method on the Writing (WRT), Roleplay (RPL), and Humanities (HUM) categories of the MT-Bench dataset (Zheng et al., 2023), all of which consist of open-ended generation tasks. Each category consists of 10 tasks, with each task comprising two turns: an initial question and a follow-up question, resulting in a total of 60 questions. We employed claude-v1, gpt-3.5-turbo, and vicuna-13b-v1.3 as target LLMs because their performance levels are relatively similar.

We compared the proposed method (PROP) with the relative evaluation methods proposed by Zheng et al. (2023) and Cook et al. (2024). Zheng et al. (2023)'s method (BS) does not utilize evaluation criteria, whereas Cook et al. (2024)'s method (TICK) uses automatically generated required criteria. For BS, evaluations affected by position bias were treated as ties. In addition, to verify the effectiveness of the proposed two-stage evaluation framework, we evaluated a single-stage framework using both required and bonus criteria (SS). In SS, all required and bonus criteria were provided to the judge LLM together, without distinction. Evaluation was performed based on the overall proportion of criteria receiving a "Yes" judgment. SS and PROP used the same set of required and bonus criteria. For all methods, gpt-4o-2024-11-20 (OpenAI, 2024) was used as both the criteria-generation LLM and the judge LLM. The temperature was set to 0, and each experiment was performed three times using different random seeds.

Following prior work (Zheng et al., 2023; Cook et al., 2024), each LLM-as-a-Judge method was evaluated based on its agreement rate with human evaluations. However, the MT-Bench dataset contains varying numbers of human annotations for different combinations of questions and target LLM pairs. Therefore, rather than using the publicly available annotations, we collected new human evaluation data by obtaining three annotations for each combination of question and target LLM pair. Specifically, we recruited three human annotators with Ph.D. degrees through Prolific. The annotators were presented with responses generated by two target LLMs and asked to determine which model performed better, with the option to indicate a tie (Figure 2 in Appendix A). The inter-annotator agreement rate was 49.1%. When calculating the agreement rate between automatic and human evaluations, the three annotations were treated independently rather than aggregated by majority voting or other methods, yielding a total of 540 samples (60 questions $\times$ 3 model pairs $\times$ 3 annotators). The upper bound of the agreement rate, defined as the agreement rate of an oracle system that always selects the majority label for each question and target LLM pair, was 72.6%.

Method	WRT	RPL	HUM	Avg.
BS	32.4	53.1	**41.1**	42.2
TICK	35.7	44.3	33.7	37.9
SS	34.1	56.1	37.2	42.5
PROP	**39.4**	**56.3**	38.1	**44.6**

Table 1: Agreement Rate with Human Evaluation (%)

3.2 Result

Table 1 presents the agreement rates between each method and human evaluations, averaged over three runs with different random seeds. Avg. denotes the macro-average across the three categories. The agreement rates for each pair of target LLMs are shown in Appendix D. As shown in Table 1, PROP achieved the highest average agreement rate and outperformed TICK across all categories. The differences between PROP and TICK were statistically significant in all categories based on McNemar's tests with $\alpha = 0.05$. Furthermore, when comparing TICK and SS, SS underperformed TICK in the Writing category, but outperformed TICK in the Roleplay and Humanities categories as well as on average. These results demonstrate the effectiveness of incorporating bonus criteria in addition to required criteria.

When comparing SS and PROP, PROP achieved higher agreement rates in all categories. These results indicate that the use of bonus criteria as well as the two-stage framework that separately treats required and bonus criteria is beneficial.

To further analyze the effectiveness of the two-stage framework, we examined how often Stage 2 was invoked and how often it resolved ties from Stage 1. Across all 60 questions, three target LLM pairs, and three runs, Stage 1 produced ties in 53.3% of comparisons (288/540). Stage 2 resolved 51.7% of these ties, reducing the final tie rate to 25.7% (139/540), which is close to the 22.2% tie rate observed in human evaluations. These results indicate that required criteria alone are often insufficient to distinguish between closely matched responses, whereas bonus criteria provide complementary information for finer-grained discrimination.

4 Discussion

4.1 Evaluation of Criteria-Generation LLM

We evaluated whether the criteria-generation LLM in the proposed method generates appropriate evaluation criteria. Specifically, we conducted a human evaluation of 273 required criteria and 266 bonus criteria generated in the experiments, each of which was assessed by three human annotators using a binary appropriate/inappropriate judgment. The inter-annotator agreement rate was 81.0% (Required: 85.1%, Bonus: 76.7%). The instructions provided to the annotators are shown in Appendix A (Figure 3).

The evaluation results demonstrated that 95.6% of the required criteria were judged appropriate by the majority of annotators, and 76.9% were judged appropriate by all three annotators. For the bonus criteria, 93.6% were judged appropriate by the majority of annotators, and 64.7% were judged appropriate by all three annotators. These results indicated that the generation accuracy of bonus criteria was lower than that of required criteria. Nevertheless, the bonus criteria achieved an appropriateness rate of over 90% based on majority judgments, demonstrating that both types of criteria were generated with high accuracy.

4.2 Evaluation of Judge LLM

We evaluated whether the judge LLM in the proposed method can accurately perform binary (Yes/No) judgments for each criterion generated by the criteria-generation LLM. Specifically, three human annotators judged whether responses generated by the three target LLMs satisfied each of the 273 required criteria and 266 bonus criteria using Yes/No labels. We calculated the agreement rate between human and judge LLM judgments, with the human label determined by majority vote among the three annotators. The inter-annotator agreement rate was 64.9% (Required: 73.6%, Bonus: 56.0%). The instructions provided to the annotators are shown in Appendix A (Figure 4).

The evaluation results showed agreement rates of 82.2% for the required criteria and 75.5% for the bonus criteria. Similar to the evaluation results for the criteria-generation LLM, the agreement rate for bonus criteria was lower than that for required criteria. Nevertheless, both achieved relatively high agreement rates, indicating that the judge LLM could judge whether responses satisfied the generated criteria with a reasonable level of accuracy.

4.3 Implications for Improving LLM-as-a-Judge

In Section 4, each component of the proposed method (i.e., the criteria-generation LLM and the judge LLM) achieved high agreement rates with human evaluations. However, the agreement rate between the proposed method and human relative evaluations remained limited to 44.6%. One possible explanation is the inherent variability in human evaluations, as indicated by the inter-annotator agreement rate of 49.1%. Another possible reason is that the proposed method treats all required and bonus criteria uniformly within each stage, without considering differences in importance among individual criteria.

This issue may partly explain why PROP underperformed BS in the Humanities category, although BS directly performs holistic relative evaluation without explicitly decomposing the evaluation into individual criteria. The Humanities category consists of questions in the humanities and social sciences, which often require domain-specific knowledge to answer appropriately. Consequently, the generated evaluation criteria tend to include both criteria related to domain-specific knowledge and criteria related to the general quality of LLM outputs. Examples of the generated criteria are shown in Appendix C. While the proposed method treats all criteria uniformly, human evaluators tended to place greater importance on criteria related to domain-specific knowledge than on general output quality. This mismatch may have contributed to the discrepancy between the proposed method and human relative evaluations. Future work will extend the proposed framework to incor-

porate importance weighting of evaluation criteria in relative evaluation.

5 Conclusion

In this paper, we proposed a new two-stage LLM-as-a-Judge framework that incorporates both required and bonus criteria for relative evaluation. The proposed framework first evaluates whether responses satisfy required criteria and then performs a finer-grained comparison using the bonus criteria. Experiments on the Writing, Roleplay, and Humanities categories of MT-Bench demonstrated that the proposed method achieved the highest average agreement rate with human evaluations among the existing methods. In addition, detailed analyses showed that both the criteria-generation LLM and the judge LLM achieved high agreement rates with human evaluations.

Limitations

This study has several limitations. First, the experiments were conducted only on the Writing, Roleplay, and Humanities categories of the English-language MT-Bench dataset. Although these categories comprised open-ended generation tasks suitable for evaluating the proposed method, the effectiveness of the method on other categories, datasets, and languages remains unverified. In particular, in the Humanities category, the proposed method underperformed Zheng et al. (2023)'s method, indicating that the proposed method may be less effective for tasks in which differences in importance among evaluation criteria are large, as discussed in Section 4.3. However, performance improvements were observed in the Writing and Roleplay categories. These results indicate that evaluation performance in LLM-as-a-Judge may be further improved by incorporating importance weighting of evaluation criteria.

Second, all experiments used a single LLM, gpt-4o-2024-11-20, for both criteria generation and judgment. We selected this model primarily to enable direct comparison with TICK (Cook et al., 2024), which also used gpt-4o as the judge LLM. Furthermore, we conducted preliminary experiments with gpt-5; however, it did not support explicit temperature control, and the evaluation results demonstrated substantially greater variation across three runs than gpt-4o (standard deviation of the agreement rate: gpt-4o = 0.003, gpt-5 = 0.020). Therefore, we used gpt-4o in the main experiments to prioritize stability and reproducibility. Evaluating the proposed method with other recent judge LLMs, including Claude Sonnet and Claude Opus, remains an important direction for future work. Independently varying the criteria-generation and judge LLMs remains a direction for future work to assess the robustness of the proposed framework.

Third, the experiments included only three target LLMs: claude-v1, gpt-3.5-turbo, and vicuna-13b-v1.3. Further experiments with a wider range of target LLMs are needed to validate the generality of the proposed method.

Ethical Considerations

We employed publicly available benchmarks (MT-Bench: Apache License 2.0) and APIs in accordance with their respective licenses, terms of use, and intended research purposes.

We reviewed all datasets used in this study for personally identifiable information and offensive content. The collected human evaluation data contained no personally identifiable information and were stored and managed in an anonymized form. The human evaluation data were collected via Prolific. Participants were informed of the research purpose prior to participation, and informed consent was obtained voluntarily.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024. Humans or LLMs as the Judge? A Study on Judgement Bias. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8301–8327, Miami, Florida, USA. Association for Computational Linguistics.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can Large Language Models Be an Alternative to Human Evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Jonathan Cook, Tim Rocktäschel, Jakob Nicolaus Foerster, Dennis Aumiller, and Alex Wang. 2024. TICKing All the Boxes: Generated Checklists Improve LLM Evaluation and Generation. In *Language Gamification - NeurIPS 2024 Workshop*.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2023. Prometheus: Inducing Evaluation Capability in Language Models. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An Open Source Language Model Specialized in Evaluating Other Language Models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

OpenAI. 2024. Hello GPT-4o. `https://openai.com/index/hello-gpt-4o/`.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024. InFoBench: Evaluating Instruction Following Ability in Large Language Models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13025–13048, Bangkok, Thailand. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. Large Language Models are not Fair Evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

A Human Evaluation Forms

Figure 2 shows the annotation form used for human relative evaluation. Annotators were presented with responses generated by two target LLMs and asked to determine which response was better, with the option to indicate a tie. Figure 3 shows the annotation form used to evaluate the appropriateness of the generated required and bonus criteria. Figure 4 shows the annotation form used to evaluate the binary (Yes/No) judgments made by the judge LLM.

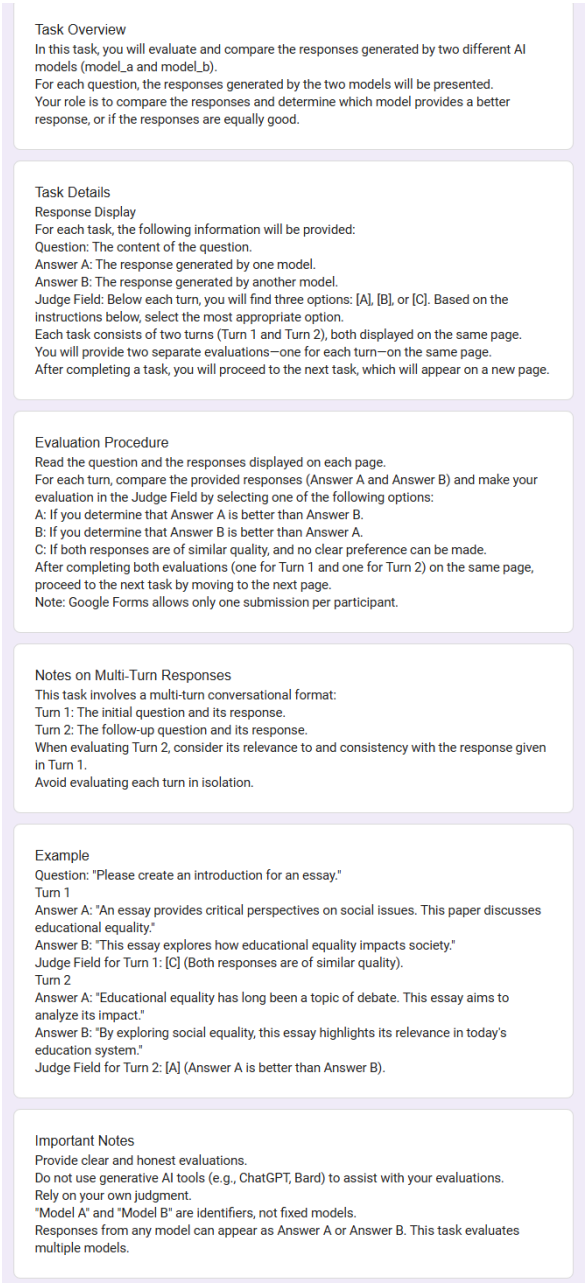


Figure 2: Annotation Form for Human Relative Evaluation

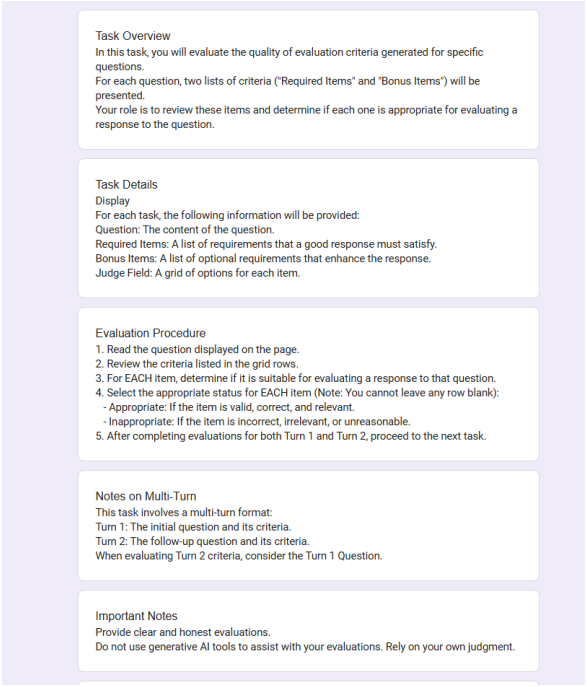


Figure 3: Annotation Form for Criteria Appropriateness Evaluation

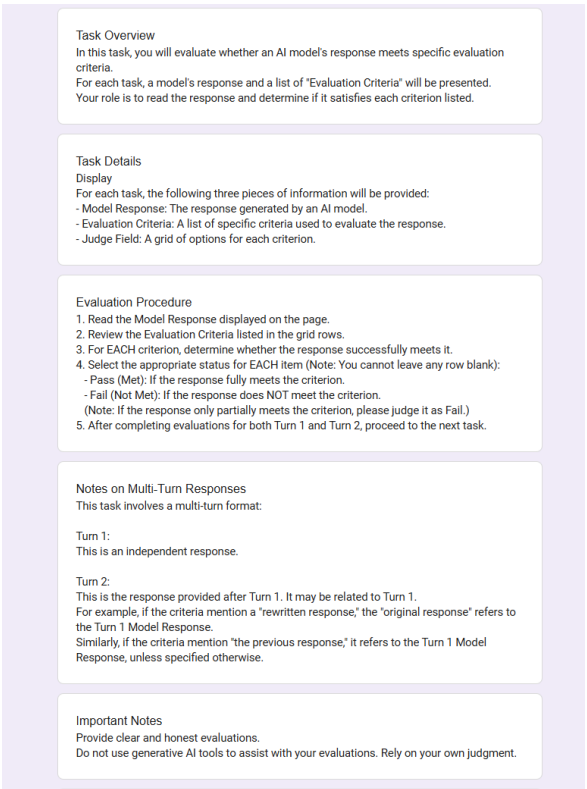


Figure 4: Annotation Form for Evaluating Binary Judgments by the Judge LLM

B Prompt

In this appendix, we present the prompts used by the proposed method for the initial question (Turn 1) in the experiments. The prompts for the follow-up question (Turn 2) are identical to those for Turn 1, except that the initial question and the model responses to the initial question are additionally provided as inputs. All prompts, including those for Turn 2, are publicly available at https://github.com/Tsunefuji/Two-Stage-LLM-as-a-Judge-paper.

Prompt for Generating Required Criteria

```
Please help judge an AI assistant's response to an instruction by providing an evaluation checklist.
To write a specific evaluation checklist, you get given the following entity each time:
INSTRUCTION: An instruction that has been given to an AI assistant.

## Task Details
Your task is to come up with an evaluation checklist list for a given INSTRUCTION.
This evaluation checklist should be a list of questions that ask whether or not specific criteria relevant to the
INSTRUCTION were met by an AI assistant's response.
Criteria covered by your checklist could be explicitly stated in the INSTRUCTION, or be generally sensible criteria for
the problem domain.
You should, however, try to be concise and not include unnecessary entries in your checklist.

Checklist questions should:
- **Be answerable by 'yes' or 'no'**, with 'yes' meaning that the response successfully met the corresponding
requirement.
- **Be comprehensive, but concise**, meaning that all criteria directly relevant to the INSTRUCTION should be
represented by a quesion, but only questions that are very clearly relevant should be included.
- **Be precise**, meaning that checklist questions should avoid vague wording and evaluate specific aspects of a
response, directly using the phrasing of the INSTRUCTION where appropriate.

You should always analyse the INSTRUCTION before providing an evaluation checklist.

## Response Format
Analysis: xxx
Answer: CHECKLIST QUESTIONS (each question should appear on a new line)

## Examples
{example}

## Real Task

### INSTRUCTION
{message}

### Response
Please analyse the instruction and provide an answer in the correct format.
Remember that each question should be phrased such that answering with 'yes' would mean that the response
**successfully** fulfilled the criteria being assessed by the question.
Your checklist should contain at least two questions, but no more than eight.
Follow the exact format shown in the examples above.
```

Prompt for Generating Bonus Criteria

```
Please create evaluation criteria for assessing responses to instructions given to an AI assistant.
To create specific evaluation criteria, the following entities will be provided each time:
Instruction: The instruction given to the AI assistant.
Required Items: The already created required items.

## Task Details
Your task is to analyze the given instruction and required items, and create bonus items for evaluating responses to
that instruction.
These bonus items should be a list of questions asking whether the quality of the response is further enhanced after
satisfying all required items.
Bonus items must not overlap in content with required items.
The criteria covered by your bonus items should be those that enhance the quality of the response.
However, you should strive for conciseness and avoid including unnecessary items.
The questions in the bonus items must be:
- Answerable with "yes" or "no". "Yes" means the response successfully satisfied the corresponding requirement.
- Comprehensive yet concise. List all criteria beyond required items that enhance response quality.
- Created avoiding ambiguous language as much as possible.

## Response Format
Make the output format exactly match the following form (including symbols and line breaks).
Analysis: While considering the required items, describe the elements that should be treated as bonus points.
Answer: Bonus Items
- (List bonus items one per line)

## Actual Task
### Instruction
{task}
### Required Items
{required_items}
### Response
Analyze the instruction and required items, and provide your answer in the correct format.
Remember that each question should be written such that answering "yes" means the response successfully satisfied the
criteria being evaluated.
```

Prompt for Judge LLM

```
Please act as a fair judge. Based on the provided Instruction and Generated Text, analyse the Generated Text and
answer the Questions that follow with 'YES' or 'NO'.
Your selection should be based on your judgment as well as the following rules:

- YES: Select 'YES' if the generated text entirely fulfills the condition specified in the question. However,
note that even minor inaccuracies exclude the text from receiving a 'YES' rating. As an illustration, consider a
question that asks, "Does each sentence in the generated text use a second person?" If even one sentence does
not use the second person, the answer should NOT be 'YES'. To qualify for a 'YES' rating, the generated text
must be entirely accurate and relevant to the question.

- NO: Opt for 'NO' if the generated text fails to meet the question's requirements or provides no information
that could be utilized to answer the question. For instance, if the question asks, 'Is the second sentence in
the generated text a compound sentence?' and the generated text only has one sentence, it offers no relevant
information to answer the question. Consequently, the answer should be 'NO'.

## Output Format
Analysis: xxx
Answer: YES / NO (this should be either 'YES' or 'NO')

## Evaluation Information

**Instruction**
{message}

**Generated Text**
{generation}

**Question**
{question}

Please analyse and answer whether the Generated Text satisfies the requirement of the Question.
```

C Examples of Generated Evaluation Criteria

Question	Required Criteria	Bonus Criteria
[Writing Category] "Compose an engaging travel blog post about a recent trip to Hawaii, highlighting cultural experiences and must-see attractions."	• Does the response take the form of a travel blog post? • Is the response engaging and well-written? • Does the response highlight cultural experiences in Hawaii? • Does the response include must-see attractions in Hawaii? • Is the response relevant to the theme of a recent trip to Hawaii?	• Does the response include vivid and sensory descriptions? • Does the response incorporate personal anecdotes? • Does the response provide practical travel tips? • Is the response well-organized with clear headings? • Does the response convey enthusiasm and a sense of wonder?
[Roleplay Category] "Pretend yourself to be Elon Musk. Why do we need to go to Mars?"	• Does the response adopt Elon Musk's speaking style? • Does the response address "Why do we need to go to Mars?"? • Does the response reflect Elon Musk's perspective and reasoning?	• Does the response include visionary or inspirational language? • Does the response provide detailed and compelling arguments? • Does the response emphasize broader implications for humanity? • Does the response incorporate references to SpaceX?
[Humanities Category] "Provide insights into the correlation between economic indicators such as GDP, inflation, and unemployment rates."	• Does the response provide insights into the correlation between GDP, inflation, and unemployment rates? • Does the response explain how fiscal policies affect the indicators? • Does the response explain how monetary policies affect the indicators? • Is the explanation clear and logically structured? • Does the response avoid factual inaccuracies?	• Does the response include real-world examples? • Does the response address potential limitations or exceptions? • Does the response provide actionable insights? • Does the response use concise language and avoid technical jargon? • Does the response highlight recent trends or developments?

Table 2: Examples of Required and Bonus Criteria Generated by the Criteria-Generation LLM for Each Category

D Results by Target LLM Pair

Table 3 presents the agreement rates with human evaluations for each target LLM pair, averaged over three runs using different random seeds.

Method	claude-v1 vs. gpt-3.5-turbo	claude-v1 vs. vicuna-13b-v1.3	gpt-3.5-turbo vs. vicuna-13b-v1.3
BS	37.0	50.6	39.1
TICK	32.8	44.8	36.1
SS	41.8	46.5	39.1
PROP	43.7	50.0	40.2

Table 3: Agreement Rates with Human Evaluations by Target LLM Pair (%)